(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
31 May 2001 (31.05.2001)

PCT

(10) International Publication Number
WO 01/39470 A1

(51) International Patent Classification[7]: H04L 29/06, 29/12

(21) International Application Number: PCT/US00/31990

(22) International Filing Date:
21 November 2000 (21.11.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/167,123 23 November 1999 (23.11.1999) US
09/ 20 November 2000 (20.11.2000) US

(71) Applicant: INFOLIBRIA, Inc. [US/US]; Suite 323, 411 Waverly Oaks Road, Waltham, MA 02145 (US).

(72) Inventors: GLINES, Stephen; 62 Tobey Road, Belmont, MA 02478 (US). LOVERSO, John, R.; 71 Pine Hill Road, Southborough, MA 01772 (US).

(74) Agents: THIBODEAU, David, J., Jr. et al.; Hamilton, Brook, Smith & Reynolds, P.C., Two Militia Drive, Lexington, MA 02421 (US).

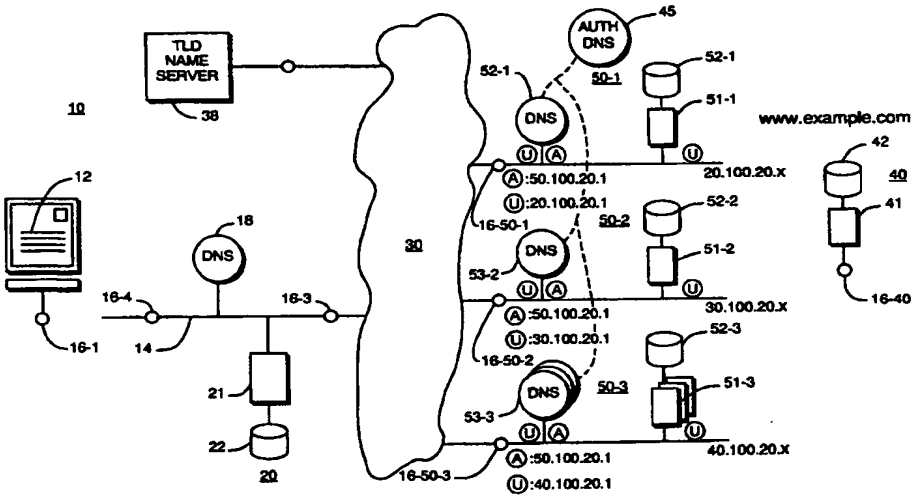
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*
— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

[Continued on next page]

(54) Title: OPTIMAL REQUEST ROUTING BY EXPLOITING PACKET ROUTERS TOPOLOGY INFORMATION



(57) Abstract: A technique for redirecting client computer requests for content files to the closest replica of the requested content, by using anycast messaging. The request to resolve a domain name is forwarded as an anycast message to a name service provided by a group of name servers distributed in the network. The closest name server then responds to the anycast message by returning a unique network address for an associated content server that contains a replica of the requested content file. This scheme permits a client computer to subsequently establish higher level protocol access method, such as a Hyper Text Transfer Protocol (HTTP) request, to open a connection and deliver the content file replica, from a content server that is topologically close to the client, using only standard network protocols.

WO 01/39470 A1

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

OPTIMAL REQUEST ROUTING BY EXPLOITING PACKET ROUTERS TOPOLOGY INFORMATION

BACKGROUND OF THE INVENTION

It has been recognized for quite some time that as computer networks grow in size, the demand for popular services grows even faster than the physical network infrastructure. It is now quite common in the Internet for individual servers and network links to become swamped with the volume of demands that are presented on a global basis.

For one common type of Internet access, namely requests originating at client browser programs for the delivery of web based content files, the process follows a fairly predictable sequence. In particular, an end user types in or clicks on a Uniform Resource Locator (URL) which consists of a fully qualified domain name, an optional directory and port number, and an access method. The client browser then makes a request to discover an Internet Protocol (IP) layer address of the web file server associated with the domain name. This domain name request message is typically submitted to a network service known as the Domain Name Service (DNS). When the correct IP address has been delivered by the DNS, the browser then attempts to open a connection to the remote file server indicated by the IP address. If the connection is successfully opened, the remote file server complies with the request and delivers the requested content file or files.

This scenario works reasonably well when the network is not saturated and when the remote system is capable of handling requests as they are presented in a timely fashion. However, the need to support high volume traffic at web sites presents a number of difficulties that must be overcome if the servers themselves are to remain viable. For example, if the number of requests for content files (i.e. "hits") exceeds the ability of a specific server to handle the demand, the server response time may become intolerably slow or the server may even crash. Other

problems occur when the nature of the content strains the network delivery infrastructure to capacity. For example, streaming video content files can easily swamp most available data conduits.

One solution to these problems, which has been known for some time, is the use of caching and replication. By replicating copies of popular content and locating the copies throughout various locations in the network, the demand on the orginal server can be offloaded. The problem then becomes one of distributing service requests among multiple servers in a fashion that minimizes network traffic.

A caching server replicates the services and content of an original or primary server. The introduction of caching can therefore markedly increase the apparent bandwidth of the original server, and provide redundancy for the original server. If the cache is topologically close to the client, then overall amount of network traffic is reduced as well.

In the most commonly thought of case, the user of a client computer may demand delivery of a content file associated with a particular URL. The speed of delivery is determined by the ability of the remote host system to deliver the data and the ability of the network to transmit it. Internet Service Providers (ISPs) that host mostly client computers can therefore improve client response by placing caching servers behind their routers. By caching the most popularly requested web sites (such as yahoo.com), an ISP can therefore significantly reduce its need for external network traffic.

Other types of systems support primarily content providers as opposed to end users. These services, such as the Internet Data Center service provided by Exodus Communications, Inc. of Santa Clara, California, maintain multiple sites with high bandwidth connections and multiple server peering arrangements. The demand for bandwidth may become extremely large in such systems and may still require interconnection to Internet backbones via high speed optical connections. These

multiple high bandwidth connections insure reliability and redundant capacity. Such systems may also provide caching of content, but are still subject to problems since they must also maintain redundant facilities as necessary to handle peak demand.

Guyton, J. D. and Schwartz, M. F. in "Locating Nearby Copies of Replicated Internet Servers," Technical Report CU-CS-762-95, Department of Computer Science, University of Colorado, Boulder, Colorado (February 1995), recognize a number of different criteria to be considered when determining how to efficiently route requests to replicated servers (also known as mirrors). These considerations involve determining whether server location information is gathered in response to specific requests or gathered proactively, in advance. Other choices involve determining whether caching support should be provided by the routing layer or the application layer. Further considerations involve the cost of polling routing tables versus gathering information via network probes or measurement beacons.

The Guyton paper also recognizes that a messaging technique known as anycast might be useful in locating cache copies. In particular, it is stated that an anycast message service is useful in situations where it is necessary to locate a host which supports a particular service, and where several servers may support the service.

Another solution which has emerged is to place many caching servers throughout the Internet. This solution spreads the load out over distant segments of the network in a manner that improves availability, while reducing the demand on any one machine. In such systems, specific algorithms are required to allow the DNS to match the client with its logically nearest cache server. In this model, requests for services are sent to a domain naming system that maintains a list of discoverable routes. Requests are then routed to the IP address of the caching system located as close to the requester as possible.

Unfortunately, this solution requires constant reevaluation of the Internet topology, such as through probing of routing tables, in order to build valid maps of the ever-changing interconnection topology. A further problem exists with the approach in that the Internet consists of many thousands of autonomous routing systems, not all of which are willing to cooperate with each other when presented with route probing requests. Other problems exist because the routes over which messages travel are not always symmetrical. Dead links or route flaps also pose a difficulty, since they cannot necessarily be discovered in real time. As a result, there will always be some percentage of dropped service requests with this approach.

Current versions of the Internet Protocol (IP) supports three types of addressing: unicasting, multicasting and anycasting. Unicasting is the most common form of addressing. In the unicast address space, every interface to the Internet has a separate IP layer address. Most machines have only one interface, although other machines such as routers may have many such addresses. Multicast messaging allows a single message to be addressed to a group of systems. It is a form of broadcast messaging in which a user may send a message to all listening recipients. Anycast messaging assumes there are multiple systems providing an identical service. Since all the machines in the anycast address space are identical, they can have the same IP address. Routing policies within the Internet Protocol can be depended upon to automatically route packets to the nearest anycast system, namely the one with the best distance metric (e.g. the least number of hops) from the client (at the current time or via the current routing topology).

A recent Internet Engineering Task Force (IETF) proposal by Katalone, G. and Rockell, R., (June 1999) recognizes that DNS servers have long suffered from availability and reachability issues. To that end, their proposal suggests a technique to maintain the availability and reachability of such an essential service. The idea is to assign the same anycast IP address to several DNS servers in a network. This provides a highly available and reliable DNS service without regard to customer or server location. All the DNS servers advertise the same unicast address, allowing

the underling routing protocol to route requests to the closest available DNS server. The anycast routing protocol itself therefore decides which DNS server machine will be used at any given point in the network. In the event that a particular DNS server becomes unavailable, that server's routing information is withdrawn from the network by the anycast routing protocol, and a new route is chosen. This technique therefore provides for high level of reachability and reliability through redundancy within the DNS system itself.

SUMMARY OF THE INVENTION

It has therefore been recognized that an anycast messaging construct can be used to locate one of several DNS resolvers in a network. However, there remain certain problems with such an approach. In particular, standard network protocols only guarantee that the route which an anycast message takes from its source to its destination is the best one at a given instant in time. Thus, two packets sent to the same anycast IP address, even if sent in sequence, one immediately after the other, may end up at two physically different anycast servers. If an anycast message comprises more than one packet, its component parts may thus actually take entirely different routes between a sender and receiver.

For example, assume a client issues an anycast request to a service that returns a reply which requires two packets at the TCP layer. Furthermore, assume the request was served by a first server and that the client's TCP stack has sent two acknowledgment packets. Because the acknowledgments are sent to an anycast address, routers may actually deliver them to different servers, causing the first server to constantly retransmit the second packet.

What is needed is a way to provide for increased performance in the delivery of requested content files in networked computer environments. The technique should have minimal impact on existing network infrastructure and require as little reprogramming and rearranging of infrastructure such as routers and gateways as

possible. In addition, the technique should not require alteration of standard network communication protocols.

In a system operating in accordance with the invention, content distribution is provided as a service where client requests are automatically routed to the closest available content server. The performance of the system as a whole may be increased by simply adding more server resources to the parts of the network that need it, with minimal change or no change to the network existing infrastructure.

To permit client requests to be rooted properly, a group of name servers are located throughout the network. The name servers are addressable via a common anycast address, as well as being individually addressable via a unicast address unique to each such name server. Each name server is peered with a nearby content server, cache server, or other file server that contains (or can serve) desired content file replicas via unicast addressing. Each name server also advertises itself as being authoritative for the domains associated with the content files stored (or to be served from) the associated content server.

A request to resolve a domain name originates as a datagram addressed to the common anycast address. The name server responding to the anycast returns an Internet Protocol (IP) address which is the unique public unicast address for its associated content server that contains and/or is able to honor the request for the specified content file. Subsequent messages needed to transfer the content file to the client can then use this unique address, and be assured that the transaction will take place with a known machine.

The following sequence of events may occur with a process implemented according to the invention. First, a user indicates the location of a content file such as by specifying a Uniform Resource Locator (URL) to a browser program, or by clicking on a hyperlink in a displayed document which has an embedded URL. Such a URL may, for example, be "http://www.example.com/homepage.html".

The browser then makes a request to a DNS service to resolve the IP address of the domain name (e.g. "example.com") specified by the URL. This request is typically formulated as a message sent to a local address resolver. If the "example.com" domain is not located in a local domain, the local resolver will then proceed to consult public name servers that have been defined as being authoritative for the "example.com" domain. The root domain name servers defined, for example, by the Internic, may be programmed to return the common address of the group of name server/content server peers that share a common anycast address.

Having now resolved an IP address for an authoritative name server (e.g., the previously defined server group address returned) for the "example.com" domain, the browser or local resolver then sends out a DNS request as a UDP datagram to an anycast message to the group. This will now resolve the IP address of www.example.com.

More specifically, since every router that is connected to one of the name server/content server pairs is advertising a route to the group address, the UDP packet will find its way first to the server pair which is located along the shortest path from the requester.

The server pair that receives the request then responds by reporting the unique (unicast) address of the associated content replica server (or cluster address if the systems are arranged in a round robin fashion). The user's browser then now make subsequent HTTP level requests to the IP address just received, to obtain the "homepage.html" file from the content server. This content server should represent the "nearest" such server, according to whatever metric the network uses to resolve the anycast address.

After the content server is found, since each content server is addressable only by its own unique IP address, conventional network protocols can then be used for the remainder of the transaction. The unicast address is used to perform

subsequent fetching of content and for any maintenance or management actions on the system, avoiding the problems which anycast addressing alone might introduce.

In order for this process to operate properly, each physical system which shares the anycast address must perform certain functions, such as a standard but customized name server function, as well as a content server, such as a cache server. Only the name server is programmed to respond to the common anycast address of the peer content server. The content server will typically never directly respond to the common anycast address (actually, it will never receive an anycast request because it does not have an anycast address).

There is no reason that the content server must be located in the same physical location and/or enclosure as the name server, but these two logical entities may actually be the same physical machine if desired.

The invention provides a solution to the problem of route discovery by avoiding it altogether. This is accomplished by using an anycast datagram to locate name servers placed on or placed logically near replicated content files. The invention exploits the fact that implicit in the Internet routing mechanism is a concept of "nearness." In this instance, nearness in the case of an anycast message, may be whatever system the Internet routing scheme first delivers a packet to. Each name server returns the IP address of one or many (such as through round robin or other mechanisms) closely bound and topologically nearby content servers. As a result, a relatively nearby content server, from the perspective or the original requesting client is always located.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of a computer network environment in which the invention may be implemented.

Fig. 2 is a flow diagram of a process which makes use of the invention.

The foregoing and other objects, features and advantages of the invention will be apparent from the following more particular description of preferred embodiments of the invention, as illustrated in the accompanying drawings in which like reference characters refer to the same parts throughout the different views. The drawings are not necessarily to scale, emphasis instead being placed upon illustrating the principles of the invention.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

Turning attention now to Fig. 1, there is shown a computer network 10 in which a mechanism is used to redirect client computer requests for content files to the closest replica of the requested content, by using anycast messaging to a name service provided by a group of name servers distributed in the network. The network 10 may be a typical client-server distributed system, such as is now popularly implemented using personal computer platforms and web based file servers connected via an internetwork 30 such as the Internet.

More specifically, a client computer 12 runs a web browser program that enables a user to submit requests for content files that are nominally located at an origin file server 40. For example, as the user types a Uniform Resource Locator (URL) into a web browser and/or uses a pointing device, such as a mouse, to click on a hyperlink embedded in a previously viewed web page, the URL for a particular web site is specified to the browser 12.

In the standard scenario, which is well known in the prior art, if the URL is fully qualified, it will typically contain a domain name, a file name for the content file, and an access method. The client browser 12 then makes a request to discover the correct Internet Protocol (IP) address of the origin server 40 that contains the requested content file. This request is submitted to a Domain Name Service (DNS)

provided by the network 10. When the DNS returns an IP address for the origin server 40, the browser program then attempts to open a connection to the origin server 40. If all goes according to design, the origin server 40 complies with the request and delivers the requested content file.

This standard scenario works reasonably well when the network 10 is not particularly busy with traffic, and the origin server 40 is capable of handling the number of requests presented to it by the client computers 12.

As the usage of the Internet continues to grow, however, difficulties must be overcome with this standard content delivery scheme. This may occur if the number of requests exceeds the ability of a specific origin server 40 to handle the demand for content files and/or the nature of the content taxes the capability of network 10 to transfer files in a reasonable expected time.

In a network 10 according to the invention, attempts have been made to alleviate such demands by associating multiple replica content servers 50-1, 50-2, ... 50-3, with the origin server 40. In particular, the replica content servers 50 may contain replicas of one or more content files that originate at the origin server 40. These content servers 50, which may typically include so called cache servers, can eliminate the need for the origin server 40 to deal with traffic demands.

Prior art schemes may typically require the reprogramming of name services to allow the browsers 12 to locate the content servers 50, such as by reprogramming one or more Domain Name Service (DNS). However, the present invention uses a particular addressing scheme to advertise the availability of alternative or replica name servers 53-1, 53-2, 53-3 for the domain located at the origin server 40. The name servers 53-1, 53-2, 53-3 are peered with content file servers 51-1, 51-2, 51-3 that contain replica copies of files stored at the origin server 40.

In one embodiment, the name servers 53 advertise themselves as reachable at an anycast address. The internetwork 30 then itself becomes responsible for delivery of a domain name request message to a closest possible name server 53. The closest name server 53 then responds to the anycast message by returning a unique IP address for its associated content server 51. This scheme permits the browser to subsequently establish the higher level protocol access method, such as a Hyper Text Transfer Protocol (HTTP) request, to open a connection and deliver the content file. Thus, having received an initial request to resolve a domain name via the anycast message route, the browser is then subsequently redirected to the associated nearest cache server 50-1, 50-2, 50-3 that can honor the remaining expected HTTP request message sequence, without adding any unnecessary traffic to the associated domain name server 53 and/or paths in the internetwork 30 to the origin server 40.

Specific aspects of the invention can now be more particularly described. As shown in Fig. 1, the environment 10 consists of a client computing device, such as a computer 12, that is running a data file retrieval program such as a web browser. The personal computer 12 is connected through an internetwork device 16-1 to a first network segment 14. The internetwork device 16-1 may be any or any combination of modem, network interface card, router, switch, bridge, gateway, or the like. The internetwork devices 16 provide the ability for connections to be made between various computing system elements using network infrastructure such as the internetwork structure 30, which may be a corporate intranet or the Internet.

In the illustrated embodiment, the client computer 12 is connected to a local area network (LAN) 14 that consists of internetwork devices 16-3 and 16-4, which in this instance are routers. A local name service such as Domain Name Service (DNS) resolver 18, a local content host 21, and local content storage device 22 also form part of the LAN 14.

The local content server 21 may be any type of well known host computer that is adapted for efficiently storing content files on a mass storage device 22.

These content files may include web pages, multimedia files, graphics, pictures, other computer files that are suitable for network transmission using well know protocols such as the HyperText Transfer Protocol (HTTP). The client computer 12 may also make connections through the local area network 14 and router 16-3 to the Internet 30 to access files located at various other computing systems. One of these computer systems may provide a service such as a root domain name service 38. Other systems serve as the origin web server 40. The origin server 40 is similar to the local host 20, in that it consists of a file server 41 and content storage 42 as well as an internetwork device 16-40.

The replica content servers 50-1, 50-2, 50-3 store replicas of one or more of the content files that originate at the origin server 40. Each content server 50 consists therefore also of a file server 51and associated mass storage device 52. Through mechanisms that are not particularly relevant to the present invention, replicas of content files that originate at the origin server 40 are distributed and stored in the replica content servers 50. Content files may be propagated through any number of schemes to push content out to various locations in the network 10 and/or move content closer to requesting client computers 12 upon demand. The connections to accomplish this are indicated by the dashed lines shown in Fig. 1. It should be understood that these are typical network connections between the origin server 40, and replica content servers 51-1, 51-2, 51-3; however, these connections are only shown here as logical connections from the perspective of the browser user client computer 12.

Also associated with each of the content servers 50 is a respective DNS server 53. The name servers 53-1, 53-2, 53-3 are addressable via both a common anycast address as well as a unique or unicast address. As will be described in greater detail below, a DNS request may be sent to the name servers 53 as an anycast datagram. The internetwork 30 is then responsible for providing best effort delivery of the datagram to at least one, and preferably the closest one, of the machines that

accept messages for the anycast address. The replica name servers 53 have received appropriate information from an authoritative DNS 45 for the domains in server 40.

Each name server 53 and file server 51 associated with particular content server 50 are considered to be connected in a peering arrangement. That is, they operate quite closely together and, in fact, are preferably located physically near one another, such as on a common local area network segment sharing the same internetwork device
16-50-1.

Each replica name server 53 therefore actually has two IP addresses, a common anycast address which is common to all of the replica name servers 53, as well as a unique unicast address which is specific to each name server 53. Each name server 53 is considered to be an authoritative DNS resolver for domain names associated with the replica content files stored in its associated replica content server 51.

To understand more particularly how a process in accordance with the invention operates, consider now Fig. 2 in connection with Fig. 1. As a first step 100, users specify a Uniform Resource Locator (URL) to a browser program running on the
client computer 12. For example, the user may specify the URL http://www.example/homepage.html.

As a next step 102, as is typical and as is well known in the art, the browser program makes an initial attempt to resolve an address for the specified domain "example.com." For example, the browser program issues a DNS request message as a UDP datagram to a name server. In the case where the user is associated with an Internet Service Provider (ISP) operating the local area network 20, this first name request is made to a local DNS resolver 18, to determine the location of the domain "example.com". The DNS resolver 18 determines whether or not the

requested content file is available locally. For example, it determines if "example.com" is located in the local web server 20. In other configurations, the resolver 18 may even reside at the client 12.

From step 104, if the content is available locally, then the local IP address is returned to the browser program in step 106.

If however, in step 104, the domain "example.com" is not available locally, then the process proceeds to step 108.

Having failed to resolve the requested name locally, a request to resolve the location of "example.com" is then sent to a root DNS server 38 in step 108. In this case, the request to the root DNS server 38 will be recursively worked through multiple root servers associated with the Internet 30 (not shown in Fig. 1) to resolve the IP address for a DNS server authoritative for the requested domain name.

In prior art systems, the root DNS name server 38 would then return the IP address of the DNS server that is authoritative for "example.com". In the illustrated embodiment, this may take the form for example, of the four-digit address 62.104.11.12 associated with a particular origin server 40.

However, in accordance with the invention, the root DNS name server 38 has been programmed to instead return the anycast address 50.100.20.1. Specifically, the name servers 53-1, 53-2, 53-3 have been designated as being the authoritative name servers for "example.com" through previous network management level configuration information. This can be done, for example, by having the parent name server (i.e. the root name server) configured to list which name servers are configured as being authoritative for the "example.com" domain. This may also be intiated at certain times, such as when the content servers 51-1, 51-2, 51-3 are populated with content file replicas from origin server 40.

As a result, the primary name service listed by organizations responsible for maintaining the state of internetwork 30, such as the Internic, will point to this common address 50.100.20.1 of the caching servers 50, instead of the origin server 40.

In step 112, now thinking that it has resolved the IP address for the single authoritative name server for "example.com", the browser then sends out a DNS request for the IP address of "www.example.com". This request message is formulated as a UDP datagram specifying the common anycast address 50.100.20.1 returned in the previous step.

The domain name request is then sent as a UDP datagram to the anycast address. In step 114 the anycast datagram will reach one of the name servers 53-1, 53-2, and 53-3 in the group associated with IP address 50.100.20.1, the one reached first being the one closest to the requesting client 12 or DNS resolver 18.

If the initial setup of the content servers 50 is such that they are located at relatively distributed locations through the Internet 30, the number of hops and hence the distance between the client 12 or DNS resolver 18 (in the case where the client 12 has a local resolver) and each particular one of the content servers 50 will be different. For example, the name server 53-1 may appear to be five hops away, the name server 53-2 may appear to be only one hop away, and the name server 53-3 may appear to be twelve hops away. Thus, the specific server 50 that will first return with a response will be the name server 53-3, as it is the closest in terms of network hops. This result is guaranteed, since every router 16 that is connected to a respective one of the content serves 50, and participates in the standard Internet routing protocols.

As a next step 116, the name server 53 associated with this closest rate will then respond by reporting the unique IP address of its associated replica content

server 51. This address is reported as a unicast address rather than an anycast address.

In a final step 118, the browser program may now make an HTTP level request for the file "homepage.html" using the standard TCP/IP and HTTP network protocols. This final request message is sent using the unicast address for the content server 51. The requested file is then returned from the content server 51-2 that is peered with the name server 53-2 that responded as being closest to the particular client 12 at the time the anycast message was sent.

It should be understood that a number of variations may be made to the present invention without departing from its scope. For example, it is not required that the name servers 53 be machines that are physically separate from the content servers 51. Indeed, in a preferred embodiment, they are running typically on the same machine with the name server 53 being of one of the processes running on the content replica server 51.

It should also be understood that an anycast message service can be built into the internetwork 30 any of a number of known ways. An anycast message service is provided for by certain types of network protocols, such as IPv6. More commonly deployed protocols, such as IPv4, do not technically have direct support for anycast. However, such protocols can be used to create a network of service groups that each act autonomously to advertise themselves as "the" gateway into a group. Such a technique for anycast using shared root servers is described at http://www.ietf.org/internet-drafts/draft-ietf-dnsop-ohsta-shared-root-server-test-00.txt and at http://www.ietf.org/internet-drafts/draft-ietf-dnsop-hardie-shared-root-server-02.txt. One other way is to use a Border Gateway Protocol (BGP) to permit anycast to work across different types of networks; other types of mechanisms such as Open Shortest Path First (OSPF), could be used, based upon the type of network in which the invention is implemented. Note also that with the present invention, DNS is only used to map the lookup of name into a local IP address of a member of

the service group. Thus, the DNS mapping needs to be established before the routing is advertised.

The selection of routing protocol may have profound effects on the propogation and convergence of group membership changes. "Membership" in the group is contigent upon distributed routing state. In the case of deployment within a single provider, where the anycast routing is internal to that network (and transparent to the outside – the Internet), and an internal routing protocol like iBGP or OSPF is used, progation of changes should be fast. In the case of deployment across multiple providers, full fledged external BGP ("eBGP") preferably would be used. Membership changes would be effected in such a scenario when the state change propagates at least half the way towards each other member in the anycast group, and that should cover all of external routers which would tag a particular anycast advertisement as "closest." State changes would possibly take longer, but would not require flooding over the entire Internet.

What is important is that a network anycast service be provided in some way that at least permits datagrams to be sent to defined groups of machines, over the best advertised route to a destination address. The anycast service however also preferably provides functionalities such as join, withdraw, failover/fallback, and overload. Each such function should perform as follows.

*join

Once a join ("begin routing advertisement") happens, the service group begins to see requests. No convergence is needed, and as the route propagates, work can be directed at the service group.

* withdraw

For a service group to orderly shutdown its participation as a member of the anycast group, it needs to stop advertising the route for the anycast address. Convergence depends upon routing protocol. For example, if iBGP is used, this is

accomplished by using the BGP message "WITHDRAW," which the routing part of the group ('gated') would send to the nearest iBGP neighbor. This would then immediately propogate out to other iBGP neighbors and cause traffic to the service group to be directed elsewhere.

* failover/fallback

If a component of one service group fails, there are several options:

- The DNS/gated host fails

First, it is postulated that each of these server processes has watchdogs on the other, such that if:

- The DNS fails, the gated does a "withdraw"
- If gated fails, DNS has nothing to do.

But, if the host or gated dies unexpectedly, then failover needs to happen. The local routing neighbor would need to depend upon the timeout facilities of the routing protocol in order to discover the outage and force the route to be removed. During this time, clients attempting to contact the anycast address would not get a response ("black-hole").

(Assuming only the DNS/gated host failed, but that the content servers in the service group were functioning, then clients who had already resolved the service name into a local IP address would continue to get service).

- If a content server in the service group fails.

Continuing the parenthetical comment above, if the client has resolved a name into the IP address of a content server in the service group, but then that individual content server fails, the client will lose access to the content.

Solutions to this problem include:

a. provide redundancy in the service group, such that a name lookup to the anycast address returns two or more IP address in the service group. This gives the client another address to try if the content server fails.

b. provide a shorter than normal TTL on the name –>local IP address mapping, such that the client is not able to cache the local IP address of a content server for an extended period of time.

Even with these optional steps, the client would see outage until the cached IP address times out.

* overload

Requests get delivered to the DNS server at the anycast address purely by the topological closeness of the requesting clients.

However, within the service group, standard load balancing and replication techniques can also apply, such as multiple content servers (returning multiple local IP addresses to a name lookup), layer 4 switches, etc. Multiple DNS servers within the group all listening to the anycast address would also be possible.

Load balancing across service groups requires an additional mechanism. Using a relatively standard approach, the DNS server in the service group can advertise a load metric to other service groups, and it can measure the load of the local content servers. When local load reached some watermark, it can load shed to content servers in other service groups by returning their IP address to name lookups coming at the anycast address.

While this invention has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the scope of the invention encompassed by the appended claims.

For example, the routers 16-50 actually advertise the fact to the routers in 30 that they know how to locate the content at "example.com" in one hop. Although they are not actually networked in this way, they advertise the availability of such content, and therefore can be considered to fool the browser into believing that a network connection is available in one hop from each of the content servers 51 to "example.com", when in fact the distance may be many hops.

Also, in order to comply with network naming conventions, the initial anycast datagram may actually return two IP addresses for the group of content servers 50. These two addresses point to the same anycast group.

The content replicas stored by the replica content servers 51 need not have all of the particular objects for the web site that they replicate.

It can now be understood how the invention makes use of anycast messaging to locate a topologically closest content replica server, without requiring the need for extensive reprogramming of naming services or unnecessary loading of the origin content host.

CLAIMS

What is claimed is:

1. A method for locating a content file in a network of server computers comprising the steps of:

storing the content file at an origin server connected to the network, the origin server having a unique network address;

placing replica copies of the content file at two or more replica content servers, the replica content servers each having a unique network address;

assigning a common network address to two or more name servers each associated with a respective one of the replica content server; and

routing name service request messages to the common network address for the name servers, such that only one of the name servers will return the unique network address of its associated replica content server.

2. A method as in claim 1 wherein the common network address for the name servers is an anycast address.

3. A method as in claim 2 wherein the name service request is an anycast datagram.

4. A method as in claim 1 wherein the unique address for the replica content server is an Internet Protocol (IP) unicast address.

5. A method as in claim 1 further comprising the step of:

in response to a request for the content file from a requesting client, serving a replica copy of the content file from the replica content server associated with the unique network address.

6. A method as in claim 1 wherein the single name server returning the unique network address is topologically closest to the requesting client.

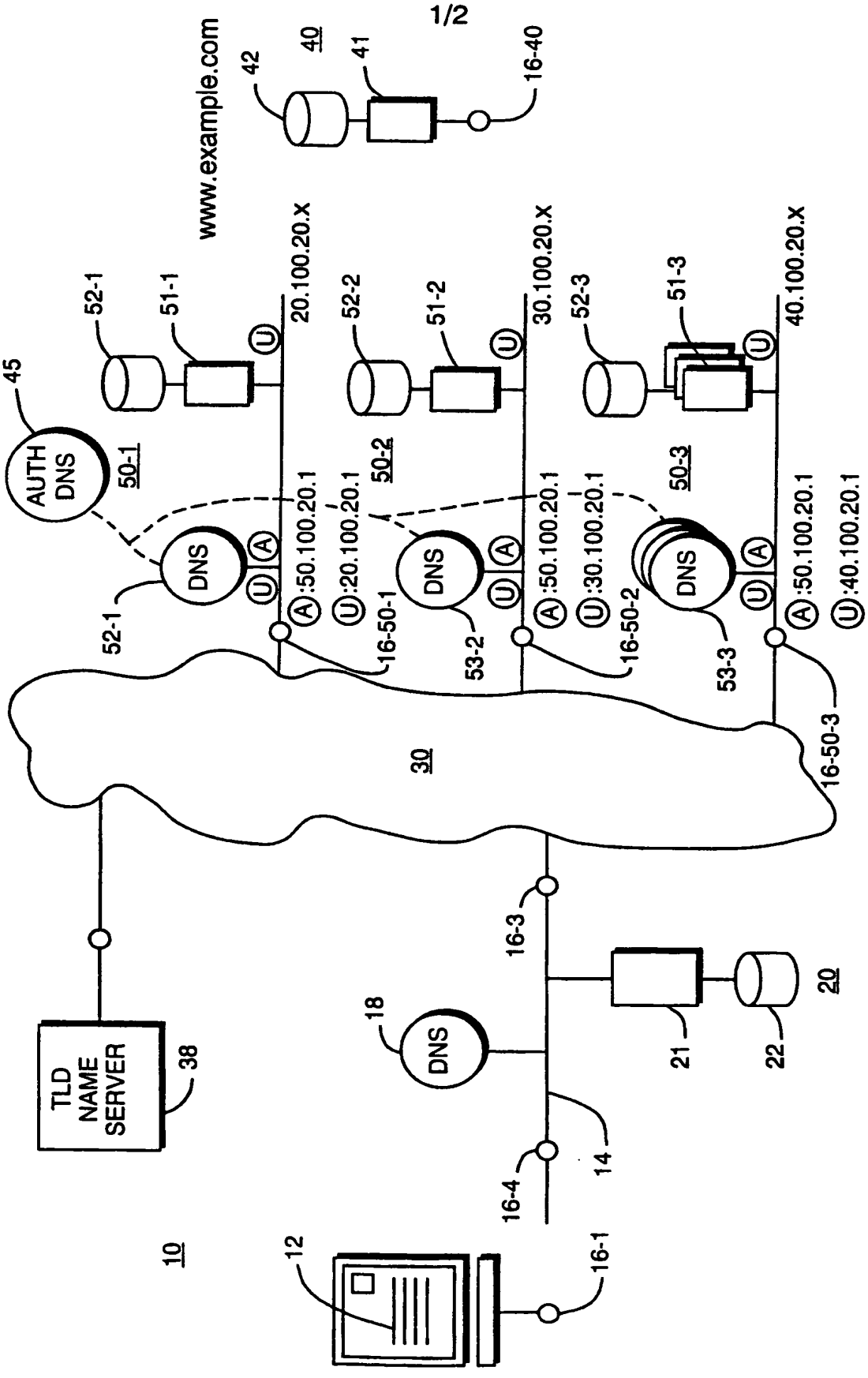

www.example.com
42
40
41
16-40
20.100.20.x
30.100.20.x
40.100.20.x
52-1
51-1
52-2
51-2
52-3
51-3
45
AUTH DNS
50-1
50-2
50-3
DNS
:50.100.20.1
:20.100.20.1
:30.100.20.1
:40.100.20.1
16-50-1
16-50-2
16-50-3
53-2
53-3
30
TLD NAME SERVER
38
18
16-3
16-4
14
12
16-1
10
21
22
20

FIG. 1

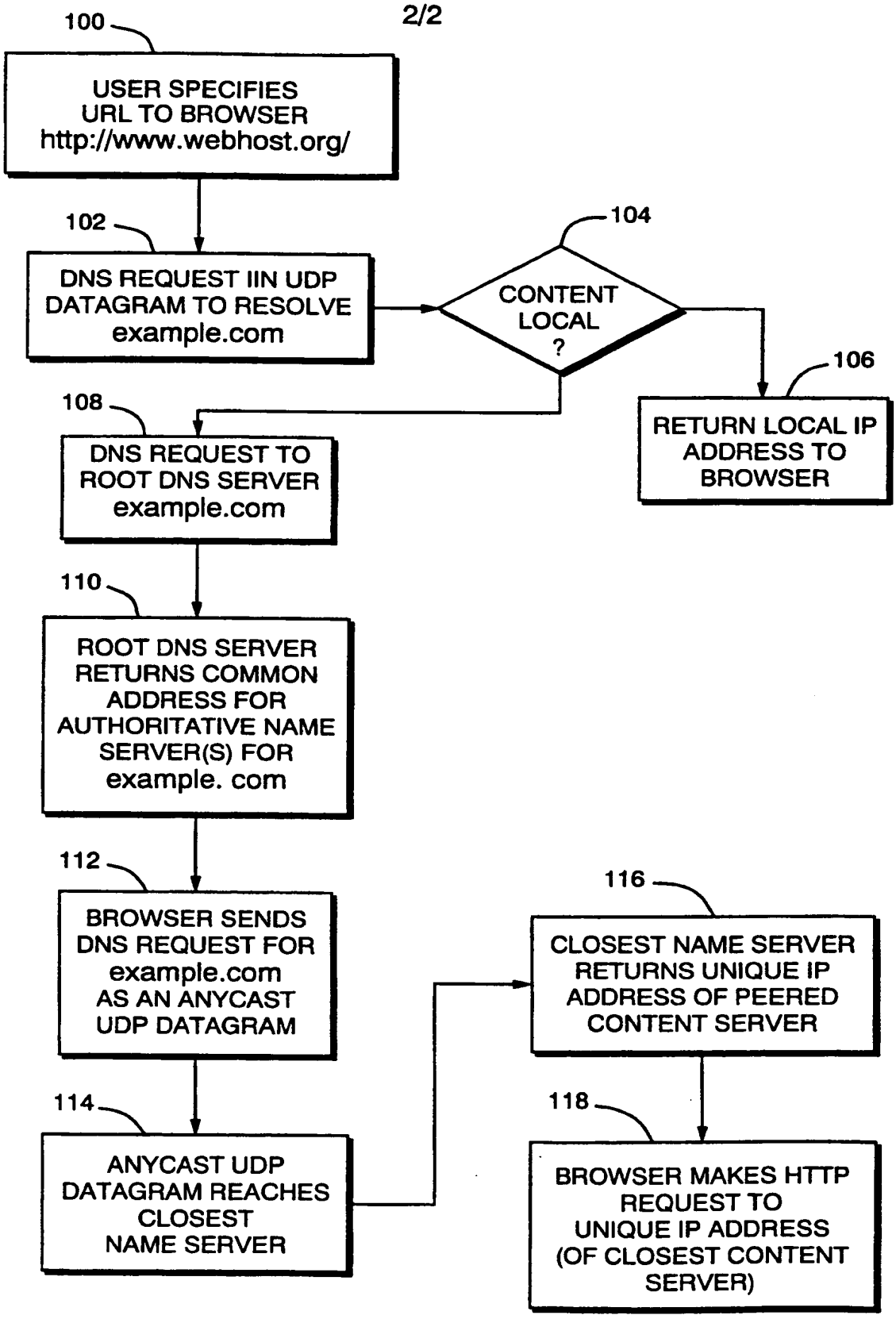

100
USER SPECIFIES URL TO BROWSER http://www.webhost.org/
102
DNS REQUEST IIN UDP DATAGRAM TO RESOLVE example.com
104
CONTENT LOCAL ?
106
RETURN LOCAL IP ADDRESS TO BROWSER
108
DNS REQUEST TO ROOT DNS SERVER example.com
110
ROOT DNS SERVER RETURNS COMMON ADDRESS FOR AUTHORITATIVE NAME SERVER(S) FOR example. com
112
BROWSER SENDS DNS REQUEST FOR example.com AS AN ANYCAST UDP DATAGRAM
114
ANYCAST UDP DATAGRAM REACHES CLOSEST NAME SERVER
116
CLOSEST NAME SERVER RETURNS UNIQUE IP ADDRESS OF PEERED CONTENT SERVER
118
BROWSER MAKES HTTP REQUEST TO UNIQUE IP ADDRESS (OF CLOSEST CONTENT SERVER)

FIG. 2

INTERNATIONAL SEARCH REPORT

Int. ional Application No
PCT/US 00/31990

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06 H04L29/12

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	BHATTACHARJEE, S.; AMMAR, M.H.; ZEGURA, ELLEN; ZONGMING FEI: "Application-Layer Anycasting" IEEE INFOCOM 1997, 'Online! vol. 3, 7 - 11 April 1997, pages 1388-1396, XP002163033 ISBN: 0-8186-7780-5 Retrieved from the Internet: <URL:http://www.ieee.org> 'retrieved on 2001-03-16! page 1388, right-hand column, paragraph 4 -page 1389, right-hand column, paragraph 1 page 1390, left-hand column, paragraph 3 -right-hand column, last paragraph page 1392, left-hand column, paragraph 1 - paragraph 2 page 1396, left-hand column, paragraph 1 --- -/--	1-6

[X] Further documents are listed in the continuation of box C. [ ] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
19 March 2001	30/03/2001
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Huber, O

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inte ..ional Application No
PCT/US 00/31990

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
Y	KATZ E D ET AL: "A scalable HTTP server: The NCSA prototype" COMPUTER NETWORKS AND ISDN SYSTEMS,NORTH HOLLAND PUBLISHING. AMSTERDAM,NL, vol. 27, no. 2, 1 November 1994 (1994-11-01), pages 155-164, XP004037986 ISSN: 0169-7552 abstract page 156, right-hand column, last paragraph -page 157, left-hand column, paragraph 1 page 157, right-hand column, paragraph 2 ---	1,4,5
A	COLAJANNI M ET AL: "ADAPTIVE TTL SCHEMES FOR LOAD BALANCING OF DISTRIBUTED WEB SERVERS" PERFORMANCE EVALUATION REVIEW,ASSOCIATION FOR COMPUTING MACHINERY, NEW YORK, NY,US, vol. 25, no. 2, 1 September 1997 (1997-09-01), pages 36-42, XP000199853 ISSN: 0163-5999 abstract page 36, paragraph 2 -page 37, paragraph 1 page 37, paragraph 4 page 38, paragraph 2 ---	1,4-6
Y	PARTIDGE C.; MENDEZ T.; MILLIKEN W.: "Host Anycasting Service" REQUEST FOR COMMENTS (RFC), 'Online! November 1993 (1993-11), pages 1-9, XP002163034 Internet Engineering Task Force (IETF) Retrieved from the Internet: <URL:http://www.ietf.org> 'retrieved on 2001-03-16! page 1, last paragraph -page 2, paragraph 4 page 3, paragraph 1 - paragraph 3 page 7, paragraph 2 - paragraph 4 -----	2,3,6

Form PCT/ISA/210 (continuation of second sheet) (July 1992)